appreciate that the timed sequence of frames 1230, 1236, 1237, 1240, 1241, 1250-1253, 1260, 1261, 1268, 1269 presented in the diagram 1200 are representative of that seen within a present day client-server electronic mail transfer.

As alluded to above, and as illustrated in native frame 1230, each of the frames 1230, 1236, 1237, 1240, 1241, 1250-1253, 1260, 1261, 1268, 1269 comprises a TCP payload field 1234, a TCP header 1233, an IP header 1232, and a MAC header 1231. In addition, since many native protocols also provide for error detection on a frame-by-frame basis, a checksum field 1235 is also depicted that contains frame error detection data so that a receiving client NIC or native port within the target adapter 1202 can detect errors that occur during transmission.

To initiate a native TCP/IP connection, the client 1220 transmits frame 1230 to the server 1210. Within frame 1230, the payload 1234 contains a request to connect to the mail server application. The TCP header 1233 contains the server TCP port number for the connection (typically a well-known TCP port number for mail transactions) and other parameters to describe the type of connection that is desired. The IP header 1232 contains the IP address of the mail server 1210 as a destination and contains the IP address of the client

1220 as a source, thus providing the server with a return IP address for IP packets. The MAC header 1231 contains source and destination MAC addresses that are modified along the path of transmission as the frame traverses the Internet from TCP/IP network to TCP/IP network. When the frame 1230 finally reaches the target adapter's local network, the MAC header will be modified to contain the destination MAC address of the target adapter 1202.

In that frame 1230 is a request to establish a connection with the server 1210, the TCP-aware target adapter 1202 embeds the frame 1230 within an IBA packet 1211 and forwards the packet 1211 to the server 1210. A connection correlation map (not shown) within the target adapter provides the DLID and work queue number for native connections with the server 1210. A connection acceleration driver (not shown) within the server 1210 receives the IBA packet 1211 and through its native transaction work queue routes the native frame 1230 up through the server's TCP/IP stack (not shown). Once the server 1210 has performed the functions corresponding to frame/packet/datagram reception within each of its MAC/IP/TCP processing layers, the connection request in payload 1234 is copied to the message reception buffer of the mail server application program. The application program, in granting the connection request, establishes a socket for communications with the client as

described above. Through this socket, the mail program sends a synchronization acknowledgement to the client granting the connection. The connection acceleration driver within the server 1210 allows this native transaction to fall through the server's TCP/IP stack and embeds the synchronization acknowledgement within IBA packet 1212, resulting in transmission of frame 1236 to the client 1220. The synchronization acknowledgement is embedded within the payload field of frame 1236. When frame 1236 is received by the client 1220, the client 1230 establishes a corresponding socket for the mail transaction with the server 1210 and generates a synchronization acknowledgement that is transmitted to the server 1210 within the payload field of frame 1237. The target adapter 1202 forwards this native frame 1237 to the server 1210 within IBA packet 1213, thus completing a three-way handshake. At this point, a TCP/IP connection has been established between the client 1220 and the server 1210.

Following establishment of the connection, the client 1220 issues a send mail request embedded as the payload of frame 1240, which is forwarded to the server 1210 in IBA packet 1214. The send mail request is processed up the TCP/IP stack of the server 1210 and provided to the mail program. The mail program receives the request and designates corresponding memory buffers that contain mail

data to be transmitted to the client 1220. IBA packet 1215 acknowledges receipt of the send mail request. The corresponding acknowledgement frame 1241 is sent to the client 1220 by the target adapter 1202.

To send the electronic mail data that is contained within the designated memory buffers to the client 1220, the application program issues a send command to the TCP layer. The send command contains a pointer to the designated memory locations. At this point, the application program waits for a notification from the TCP layer that the data has been received by the client. The connection acceleration driver intercepts this send command at the transport driver interface to the TCP/IP stack and issues an accelerated connection request to the TCP-aware target adapter 1202 in IBA packet 1216. The accelerated connection request 1216 contains TCP/IP connection parameters and memory locations of the message data, thus allowing the target adapter 1202 to map an accelerated work queue for transfer of the data. The target adapter 1202 sends IBA packet 1217 to the server 1210 granting the accelerated connection and designating the accelerated work queue number.

To transfer the data, the target adapter 1202 sends an RDMA read command in IBA packet 1218 to the server, directing a remote DMA of server memory at the memory

locations containing the message data. DMA logic within the server's host channel adapter performs the DMA and the mail data is transferred to the target adapter via packet 1219, totally bypassing the server's TCP/IP stack. And as FIGURE 12 illustrates, all of the ensuing frames 350, 351, 358, 359, 360, 361, 368, 369 that are required to deliver the data to the client 1220 are generated and/or processed by the TCP-aware target adapter 1202, completely offloading a significant amount of TCP/IP-related processing which would otherwise be required of the server 1210. Depending on the amount of data that is provided in packet 1219, this offload could result in processing savings corresponding to the generation of perhaps tens of TCP datagrams, hundreds of IP packets, and thousands of native frames 1250-1253, 1260, 1261, 1268, 1269.

As in the discussion with reference to FIGURE 3, the present discussion presents the frame structure, TCP requests, and application program commands in the timing diagram 1200 in simplified terms to illustrate the essential transactions of a server-client mail transfer according to the present invention without encumbering the reader with details associated with a specific mail server application program, operating system, or network interface. One skilled in the art will acknowledge that the transactions presented in FIGURE 12 are representative of those essential

transactions required for the transfer of electronic mail messages in virtually any present day TCP/IP-enabled mail server. Furthermore, one skilled in the art will appreciate that although the example of FIGURE 3 relates to the delivery of electronic mail messages to a client, frames 1250-1253, 1260, 1261, 1268, 1268 are indeed representative of any type of data transfer between a server and a client.

Now referring to FIGURE 13, a block diagram is presented featuring a system 1300 according to the present invention for accelerating client-server TCP/IP connections over an Infiniband Architecture network subsystem, where a TCP-aware target adapter 1330 is employed to provide TCP/IP transactions that are encapsulated within Infiniband packets over an IBA fabric to an Infiniband-to-native protocol translator 1350. The system 1300 includes one or more servers 1310 that are located within a data center 1302. The servers 1310 are interconnected over a data center point-to-point IBA fabric via Infiniband host channel adapters (HCAs) 1318. The Infiniband HCAs 1318 interface directly to a server's memory 1316 as opposed to interfacing to a CPU via a host bus 1314. The IBA fabric comprises a number of point-to-point links 1304 and cascaded switches 1320 that interconnect end nodes 1310, 1330, 1350 including host nodes 1310, a TCP-aware target adapter 1330, and a simple Infiniband-to-native protocol translator 1350. The

block diagram also depicts a number of clients 1342 that are interconnected over a TCP/IP-based client LAN 1340. Accordingly, the client LAN 1340 may employ one of the native network protocols discussed above. In this type of accelerated connection configuration, the IBA-to-native translator 1350 interfaces the Infiniband fabric to the client LAN 1340.

In operation, the elements of the system 1300 illustrated in FIGURE 13 function like elements of the system 400 discussed with reference to FIGURE 4 that have the same tens and ones digits. The difference between the system 1300 of FIGURE 13 and the system 400 of FIGURE 4 is that the system 1300 of FIGURE 13 is provided to accelerate client-server TCP/IP connections within a data center 1302 that utilizes Infiniband raw packet protocol for TCP/IP communications. Under an IBA raw packet protocol scheme, TCP/IP transaction packets are encapsulated within Infiniband packets by a sending device having TCP/IP processing capabilities. The IB-to-native translator 1350 strips IBA headers from outgoing encapsulated TCP/IP packets and routes the TCP/IP packets over the LAN 1340 to a client device 1342. The translator 1350 also encapsulates incoming TCP/IP packets from the LAN 1340 into Infiniband raw packets for transmission to a destination server 1310. The translator 1350 does not perform any TCP/IP stack functions

such as timing, flow control, etc. These functions are presumed to be performed by the servers 1310 as well. The IBA-to-native translator 1350 maintains a connection map that associates either destination MAC addresses or destination IP addresses of TCP/IP packets received from the client LAN 1340 with a corresponding DLID/WQ# pair for routing of IBA packets over the IBA fabric.

The TCP-aware target adapter 1330 of FIGURE 13 is employed to offload TCP/IP stack functions from the servers 1310 as described with reference to FIGURE 4. In addition, the target adapter 1330 also performs the functions of Infiniband packet encapsulation and stripping. Thus, in one embodiment, incoming and outgoing unaccelerated TCP/IP packets between the servers 1310 and the IB-to-native translator 1350 are routed through the target adaptor 1330. In an alternative embodiment, incoming and outgoing unaccelerated TCP/IP packets are routed directly between the servers 1310 and the translator 1350. However, when accelerated connections are established as discussed above, the target adapter 1330, in addition to performing all the TCP/IP stack functions, performs Infiniband packet encapsulation and stripping functions as well. Accelerated connections to access data in server memory are performed by the target adapter 1330 via IBA remote DMA commands as is described above. The architecture of a TCP-aware target

adapter 1330 for additionally performing Infiniband packet encapsulation/stripping functions is more specifically described with reference to FIGURE 14.

Referring to FIGURE 14, a block diagram is presented illustrating an alternative embodiment of a TCP-aware target adapter 1400 according to the present invention that is employed within the system configuration 1300 of FIGURE 13. The target adapter 1400 includes an IBA target channel adapter 1401 that is coupled to an accelerated connection processor 1430 via bus 1420. In one embodiment, both the target channel adapter 1401 and the accelerated connection processor 1430 exist as logic elements 1401, 1430 within the same integrated circuit. In an alternative embodiment, the target channel adapter 1401 and the accelerated connection processor 1430 are separate integrated circuits within the same circuit card assembly. In a PCI-based embodiment, bus 1420 is a PCI or PCI-X bus 1420. In a further alternative embodiment, the target channel adapter 1401 and the accelerated connection processor 1430 reside on different circuit card assemblies that are interconnected over a PCI/PCI-X bus 1420.

The IBA target channel adapter 1401 has a transaction switch 1402 that is coupled to a plurality of IBA MAC controllers 1414 via a corresponding plurality of

transaction queues 1418. Data is transferred between the MACs 1414 and the switch 1402 via a plurality of data buses 1416. Each MAC 1414 couples to IBA serializer-deserializer logic 1412, which provides physical interface of IBA symbols to a corresponding IBA link 1410. IBA transactions are provided to the transaction switch 1402 through each transaction queue 1418. Payload data for IBA transactions is routed via data buses 1416 to transaction data memory 1404 within the switch 1402. The transaction switch 1402 is also coupled to a protocol engine 1408 via bus 1406.

The accelerated connection processor 1430 includes a TCP/IP stack 1436 that is coupled to Infiniband packet processing logic 1460 via a plurality of native MAC logic elements 1438. The Infiniband packet processing logic 1460 encapsulates native protocol packets within Infiniband raw packets for transmission to an Infiniband-to-native translation device, like that described with reference to FIGURE 13. The Infiniband packet processing logic 1460 also strips off Infiniband headers from Infiniband raw packets received from the translation device. The TCP/IP stack 1436 is also coupled to a plurality of target protocol drivers 1434. The protocol drivers are coupled to a connection correlator 1432. In one embodiment, the plurality of target protocol drivers 1434 and MAC logic elements 1438 provide for TCP/IP native network frame transmission and reception

in accordance with a single native network protocol. In an alternative embodiment, frame processing according to two or more native protocols is provided for by the drivers 1434 and MAC elements 1438.

In operation, elements of the TCP-aware target adapter 1400 function very much like those elements described with reference to the TCP-aware target adapter 900 of FIGURE 9 that have the same tens and ones digits. The difference between the two target adapters 900, 1400, however, is that the target adapter 1400 of FIGURE 14 does not have any native protocol ports. Instead, native transactions produced by the TCP/IP stack 1436 and MAC logic elements 1438 are passed to the packet processing logic 1460, which encapsulates the native transactions within Infiniband packets. The Infiniband packets are provided to the channel adapter 1401 via bus 1420 for transmission over the IBA fabric 1410 to the Infiniband-to-native protocol translator. Even though the native transmissions are encapsulated into Infiniband packets, TCP/IP and MAC processing functions such as timing, windowing, and etc., are still performed by the TCP-aware target adapter 1400. Accelerated connections operate in the same manner as was discussed above with reference to FIGURE 9, the only difference being that Infiniband headers are added to and stripped from the transactions by the processing logic 1460.

Referring to FIGURE 15, a block diagram is presented illustrating an IB-to-native translator 1500 according to according to the present invention such as has been alluded to with reference to FIGURES 13-14. The native translator 1500 includes an IBA target channel adapter 1501 that is coupled to an unaccelerated connection processor 1570 via bus 1520. In one embodiment, both the target channel adapter 1501 and the unaccelerated connection processor 1570 exist as logic elements 1501, 1570 within the same integrated circuit. In an alternative embodiment, the target channel adapter 1501 and the unaccelerated connection processor 1570 are separate integrated circuits within the same circuit card assembly. In a PCI-based embodiment, bus 1520 is a PCI or PCI-X bus 1520. In a further alternative embodiment, the target channel adapter 1501 and the unaccelerated connection processor 1570 reside on different circuit card assemblies that are interconnected over a PCI/PCI-X bus 1520.

The IBA target channel adapter 1501 has a transaction switch 1502 that is coupled to a plurality of IBA MAC controllers 1514 via a corresponding plurality of transaction queues 1518. Data is transferred between the MACs 1514 and the switch 1502 via a plurality of data buses 1516. Each MAC 1514 couples to IBA serializer-deserializer logic 1512, which provides physical interface of IBA symbols

to a corresponding IBA link 1510. IBA transactions are provided to the transaction switch 1502 through each transaction queue 1518. Payload data for IBA transactions is routed via data buses 1516 to transaction data memory 1504 within the switch 1502. The transaction switch 1502 is also coupled to a protocol engine 1508 via bus 1506.

The unaccelerated connection processor 1570 has a native processor 1533. The native processor 1533 includes encapsulation logic 1535 and strip logic 1537. The encapsulation logic 1535 encapsulates native protocol packets within Infiniband raw packets for transmission over an IBA fabric to a server or to a TCP-aware target adapter, like those described with reference to FIGURE 13. The strip logic 1460 strips off Infiniband headers from Infiniband raw packets received from the IBA fabric for transmission of TCP/IP packets to a client over a native LAN 1550. The native processor 1533 is coupled to an unaccelerated connection correlator 1531 and to a plurality of native network ports 1540. Each of the native network ports 1540 is connected to a native client LAN 1550. In one embodiment, the plurality of native network ports 1540 provide for TCP/IP native network frame transmission and reception in accordance with a single native network protocol. In an alternative embodiment, frame processing

according to two or more native protocols is provided for by the native ports 1540.

In operation, elements of the IB-to-native translator 1500 function very much like those elements described with reference to the TCP-aware target adapter 900 of FIGURE 9 that have the same tens and ones digits. The difference between the target adapters 900 and the translator 1500 is that the translator does not have any TCP/IP-related processing logic such as a TCP/IP stack, target protocol drivers, or MAC logic elements. Instead, all TCP/IP processing functions are performed by servers or a TCP-aware target adapter connected to the IBA fabric 1510, and by client devices connected to the LAN 1550. All IBA packets received by the translator 1500 over the IBA fabric 1510 have encapsulated TCP/IP packets within. To route these TCP/IP packets to a client device, strip logic 1537 within the native processor 1533 strips out the IBA encapsulation data and formats MAC and/or IP header data according to mappings provided by the unaccelerated connection correlator 1531. The TCP/IP packets are then transmitted to the client device over one of the native network ports 1540. All TCP/IP packets received by the translator 1500 over the native LANS 1550 must be encapsulated within IBA raw packets for transmission over the IBA fabric 1510. To route these IBA raw packets to a server or to a TCP-aware target

adapter, encapsulation logic 1535 within the native processor 1533 encapsulates the TCP/IP packets into IBA raw packets and assigns destination local identifier (DLID) fields and work queue numbers within the IBA raw packets according to mappings provided by the unaccelerated connection correlator 1531. The IBA raw packets are then transmitted to a designated server or to a TCP-aware target adapter over the IBA fabric 1510.

Now referring to FIGURE 16, a block diagram 1600 is presented showing how native MAC connections are mapped within a an unaccelerated connection correlator employed by the native translator of FIGURE 15. The block diagram 1600 shows a native MAC-to-IBA map 1610 and an IBA-to-native MAC map 1620. The native MAC-to-IBA map 1610 associates destination MAC addresses that are picked from native frame headers and their payloads received from a client network with a particular destination local identifier (DLID) 1611 and corresponding work queue number 1612 for unaccelerated TCP/IP communications between the client and either a particular server or a TCP-aware target adapter connected to the IBA fabric. In a generalized MAC sharing embodiment, the native MAC-to-IBA map 1610 may be dynamically managed such that a single incoming destination MAC address is mapped to several different DLIDs/WQs 1611/1612. The IBA-to-native MAC map 1620 associates source local identifiers

and work queue numbers that are picked from incoming IBA raw packet headers received from a server/TCP-aware target adapter to a particular source MAC address 1621 which is employed within a MAC headers for native frames sent to a client. The native maps 1610, 1620 within an unaccelerated connection correlator according to the present invention allow a an IB-to-native translator to route transactions between clients connected to a TCP/IP client LAN and servers/TCP-aware target adapters connected to an IBA fabric.

Now referring to FIGURE 17, a block diagram 1700 is presented showing how native IP connections are mapped within a an unaccelerated connection correlator employed by the native translator of FIGURE 15. The block diagram 1700 shows a native IP-to-IBA map 1710 and an IBA-to-IP map 1720. The IP-to-IBA map 1710 associates destination IP addresses that are picked from native packet IP headers received from a client network with a particular destination local identifier (DLID) 1711 and corresponding work queue number 1712 for unaccelerated TCP/IP communications between the client and either a particular server or a TCP-aware target adapter connected to the IBA fabric. This mapping scheme within an unaccelerated connection correlator can be used in a load sharing embodiment of the system of FIGURE 13, a firewall embodiment, an IP security embodiment, or any other

embodiment where it is important to select a DLID/WQ# based upon destination IP address in a received TCP/IP translation. The IBA-to-IP map 1720 associates source local identifiers and work queue numbers that are picked from incoming IBA raw packet headers received from a server/TCP-aware target adapter to a particular source MAC address 1721 and source IP address 1722 which are employed within MAC headers and IP headers for native frames sent to a client.

Now referring to FIGURE 18, a block diagram is presented featuring a system 1800 according to the present invention for accelerating client-server TCP/IP connections over an Infiniband Architecture network subsystem, where multiple TCP-aware target adapters are employed to provide TCP/IP transactions over multiple client local area networks 1840. The configuration shown in FIGURE 18 is identical to the configuration depicted in FIGURE 4, with the exception that two TCP-aware target adapters 1830 are shown interfacing to two client networks 1840. The mapping scheme discussed with reference to FIGURES 10 and 11 supports multiple server-target configurations. In one embodiment, the multiple target adapters 1830 provide redundant paths to the same client network 1840. In an alternative embodiment, the multiple target adapters 1830 provide for fail-over routing. In a load-balancing embodiment, the multiple target adapters 1530 support a balanced provision of

services from a multiple servers 1810 within the data center 1802. Although only two TCP-aware target adapters 1530 and client LANS 1540 are depicted in FIGURE 15, one skilled in the art will appreciate that the accelerated connection mapping scheme according to the present invention will support a many server 1810-to-many target adapter 1830 configuration as well. In an alternative embodiment, one or more the TCP-aware target adapters 1830 within the system 1800 of FIGURE 18 can be replaced by a combination of an IB-to-native translator and target adapter having only IBA ports as described with reference to FIGURES 13-17.

The present overcomes the notable amount of TCP/IP/MAC-related processing that servers must perform in order to accomplish transfer of service result data to a client by employing IBA apparatus and method to offload this processing to a target adapter. In accordance with the present invention, the number of servers within a data center can be scaled without impacting hardware or software corresponding to the client network. Furthermore, the technology of client networks can be upgraded without impacting servers within an existing data center.

Although the present invention and its objects, features, and advantages have been described in detail, other embodiments are contemplated by the present invention

as well. For example, the present invention has been particularly characterized in the context of web page servers within a large data center. Although web page services today account for a large majority of the services provided over TCP/IP networks, other types of server applications are anticipated as well. Such services include remote TCP/IP-based storage services and file distribution. The present invention is exceptionally well suited to offload TCP/IP processing for streaming media servers, voice over IP (VoIP) communications, and sects of the industry where the movement of large amounts of data is time constrained.

In addition, the present invention has been described in terms of a connection acceleration driver that exists within server memory in a fashion that circumvents an existing TCP/IP stack within the server's operating system. And although this type of interface is anticipated in the near term, as server architectures migrate to the point where TCP/IP is no longer part of the operating system, the present invention contemplates a connection acceleration driver having an integral TCP/IP stack, very much like that provided within a TCP-aware target adapter according to the present invention. Use of this type of embodiment allows a server that does not have TCP/IP capability (i.e., perhaps Infiniband only) to execute legacy TCP-based application

programs that provide connectivity to TCP/IP-based client networks.

Furthermore, the present invention has been described as providing for both native and accelerated TCP/IP connections in a number of native protocols that are presently employed today such as Ethernet, FDDI, etc. But native protocols evolve, as seen in the case of emerging Gigabit Ethernet technologies. Application of the present invention comprehends this evolution of native protocol technologies by allowing the native protocol of a network to be upgraded in such a manner that the commensurate changes to servers in a data center are minimized to perhaps upload of driver software.

Moreover, the present invention contemplates offload of the processing require of a server to move application data. The present inventors view this type of embodiment as one that will predominately be employed. However, the architecture of the present invention also supports connection acceleration at the TCP level. In a TCP-accelerated embodiment, TCP segments are retrieved from the server as opposed to application data. Such an embodiment has sees application in certain types of servers that do not provide for direct access of some applications, perhaps for security reasons.

Those skilled in the art should appreciate that they can readily use the disclosed conception and specific embodiments as a basis for designing or modifying other structures for carrying out the same purposes of the present invention without departing from the spirit and scope of the invention as defined by the appended claims.

What is claimed is: